

A Graphical Operator Interface for a Telerobotic Inspection System

Won S. Kim, Kam S. Tso, and Samad Hayati
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109

Abstract

Operator interface has recently emerged as an important element for efficient and safe operator interactions with the telerobotic system. Recent advances in graphical user interface (GUI) and graphics/video merging technologies enable development of more efficient, flexible operator interfaces. This paper describes an advanced graphical operator interface newly developed for a remote surface inspection system at Jet Propulsion Laboratory. The interface has been designed so that remote surface inspection can be performed by a single operator with an integrated robot control and image inspection capability. It supports three inspection strategies of teleoperated human visual inspection, human visual inspection with automated scanning, and machine-vision-based automated inspection. New features implemented include extended automated sequencing, automated scanning with operator-interactive object registration and scan path graphics overlay, effective flaw notification with flaw outline graphics overlay, and on-line flaw manipulation allowing direct operation on captured video images.

1 Introduction

Space platforms such as the Space Station Freedom are planned to be operational for a long period of time in harsh low-Earth-orbit space environments. It is thus anticipated that they require both periodic and contingent inspections [2] to detect and examine surface damage, for example, impacts by space debris or by micro-meteorites [14]. Since EVA (extravehicular activity) astronaut time will be very limited and costly, there is a genuinely growing interest in both on-board and ground-controlled remote surface inspection using robot arms.

Telerobotic inspection operations, however, in general require a large number of parameter settings and monitoring capabilities, involving both remote robot control and image inspection, and thus a simple hardware-oriented one-sensor/one-display interface becomes inefficient, often demanding more than one highly experienced operator during the operation. Recent advances in graphics and graphical user interface (GUI) technologies enable us to develop a graphical operator interface design [8] for increased efficiency and safety. Graphics display and graphics/video merging (graphics overlay on video images) technologies [9], [10] provide an efficient means of pictorial communications between the human operator and the system. Graphical user interface technologies [1], [5], [11], [13] using graphics-oriented software such as windows, menus, and icons enable a more sophisticated flexible operator interface. Many pioneering graphical user interface ideas were developed or refined at Xerox Palo Alto Research Center (PARC) in 1970's [15]. Later, in 1984, Apple introduced the Macintosh desk-top computer with an easy-to-use graphical user interface with great commercial success. Since then, more than dozens of different graphical user interfaces and development toolkits have been developed for commercial market share.

This paper describes a new efficient operator interface design for a telerobotic surface inspection system [3], [4] developed at the JPL Telerobotic Surface Inspection Laboratory (JPL). After a brief description of the JPL telerobotic inspection system in Section 2, details of our graphical operator interface design supporting three complementary inspection strategies are described in Section 3, and an interactive flaw manipulation capability of our interface design is described in Section 4. Finally, conclusion and future plans are presented in Section 5.

2 Telerobotic Surface Inspection System

A telerobotic surface inspection system [3], [4] has been newly developed at the Jet Propulsion Laboratory (JPL) to demonstrate and evaluate its capabilities for potential applications in space platforms such as the Space Station Freedom. In the present ground-based demonstration (Fig. 1), the JPL telerobotic inspection system uses a 7-degree-of-freedom (dof) Robotics Research Corporation (RRC) arm mounted on a 1-dof mobile platform. The arm carries inspection cameras, controlled lights, and other sensors for inspection and manipulation. Two VME chassis are used for real-time software: one for manipulator control and the other for automated image inspection. For a simulated inspection task mockup, orbital replacement units (ORU's) are mounted on a one-third scale mock-up of a truss structure of the Space Station Freedom. More detailed general descriptions of the JPL telerobotic inspection system can be found in [3], [4].

The operator stays in the local-site operator control station and interacts with the remote-site inspection system through the operator interface. The operator control station is housed in a Space Station cupola mockup to realistically simulate the equipment and operator space limitations (Fig. 2). The operator interface hardware consists of three high resolution color video monitors, a Silicon Graphics IRIS 4D/70-GT workstation, and two 3-dof (one 3-dof translational and one 3-dof rotational) hand controllers that are mechanically identical to the ones used in the Space Shuttle Remote Manipulator System (RMS). The Shuttle-RMS-type hand controllers were selected to serve as a "standard" input device for teleoperated control, so that other input devices can be later compared with this standard device in terms of teleoperation performance. The operator interface software resides in the Silicon Graphics IRIS workstation. It consists of communication interfaces to the remote-site manipulator control and image inspection systems, on-line graphics visualization, and graphical user interfaces (GUI's). Currently, communication to the remote-site system is based on Internet domain TCP/IP sockets. In this paper, new developments of the graphical operator interface for the JPL telerobotic inspection system are described.

3 Operator Interface Design

The operator interface software was all written in C using X, Motif, Wcl, GL, and GLX. The X window system [18] is an industry standard software enabling development of "device-independent" portable GUI's. The Motif widget set [12], [18] is an industry-prevailing X

toolkit that provides useful widgets such as buttons, menus, and scroll bars. The Wcl Widget Creation Library [16] enables X resource files to specify a widget hierarchy (parent-child relationship tree), widget types, and bindings of callbacks. Since widget creations can be conveniently defined in extended X resource files, the use of Wcl can greatly reduce the programming effort. GL is the Silicon Graphics standard graphics library for graphics rendering. GLX allows to create a special window that accepts both X and GL functions for the Silicon Graphics workstation.

Following a general GUI design principle [1], [7], [13], the operator interface for the JPL telerobotic surface inspection system is hierarchically structured by grouping similar functions together. Major and commonly used functions are presented at the top-level, and less frequently used functions are hidden in the lower levels. The top-level screen layout (Fig. 3) of our operator interface design is composed of user reconfigurable multiple windows providing top-level GUI's for robot control, light control and image operations, video switch, auto sequencing, display and control sliders, automated scanning, and automated inspection.

The overall design goal of the operator interface for the JPL remote surface inspection system is to provide an efficient single-operator interface with an integrated manipulator control and image inspection capability. The interface supports three complementary inspection strategies: teleoperated (manual-scan) human visual inspection, human visual inspection with automated scanning, and machine-vision-based automated inspection. The teleoperated human visual inspection is the baseline inspection mode. When the automated inspection notifies the operator of a potential flaw occurrence, the operator may temporarily use the teleoperation mode for fine examination. The operator can also use the teleoperation mode as a back-up mode, when automated scanning or automated inspection fails.

3.1 Teleoperated Human Visual Inspection

In the teleoperated human visual inspection, the operator inspects the object surface visually through video monitors, while manually scanning the surface by looking at video monitors and teleoperating the robot arm carrying inspection cameras and controlled lights. When a flaw appears, the operator can stop scanning and capture the video image containing the flaw on the close-view window (bottom right window in Fig. 3) of the IRIS workstation. The operator can then further examine the flaw and save the flaw image of a

small rectangular region by clicking the flaw with a mouse. The operator can also mark the flaw location in the far-view image on the far-view window (next to the close-view window in Fig. 3) as described in more detail in Section 4.

In order to control the robot arm manually, the operator first sets an appropriate mode of teleoperation by using the robot control GUI (Fig. 4a; upper left window in Fig. 3). Control parameters include: 1) arm power on or off, 2) real arm drive or simulation, 3) joint, cartesian world, or cartesian tool motion, 4) hand controller position scale, and 5) tool length. Four auto sequence buttons (home, auto1, auto2, and auto3) are also provided for easy execution of frequently used pre-programmed automated motions. A simple example of a hierarchical design can be explained with the “hand controller scale” button. When the operator clicks on this button, a pop-up menu (Fig. 4b) shows up. The operator can then select a desired scale among the low, medium, and high default values, or can select the “set” button to specify the scale numerically using the “set scale” pop-up window (Fig. 4c). In the shared control mode, the operator can perturb or modify the automated motion by using hand controllers.

The display and control sliders GUI (lower left window in Fig. 3) displays both the currently measured and operator-commanded robot positions. This interface allows the operator to issue an operator-commanded auto-move by specifying the robot target position and traverse time interval. The target position is specified by using either operator-commanded robot position sliders or text widgets in any one of the joint, cartesian world absolute, cartesian world relative, and cartesian tool relative modes. The traverse time interval is specified either by setting the traverse time interval directly or, alternatively, by setting the motion speed (low, mid, high, and set).

The light control and image operations GUI (top, second left window in Fig. 3) allows the operator to control the light intensity levels through slider control or numerical text entry. This interface also has auto sequence buttons for image operations. The video switch GUI (below the light control GUI in Fig 3) allows the operator to connect any output channel (typically video monitors) to any input channel (typically video cameras). The operator selects the output channel first and then the input channel. Each output-channel pushbutton widget displays the currently connected input channel. Typically one monitor is used for surface inspection, and the other two for manipulator control.

An extended automated sequencing capability has been developed to support both manipulation and image inspection commands. Our implementation can be considered as an

extension of the current Shuttle RMS auto sequence capability [6] which is limited to the pre-programmed robot motion control. Further, our implementation supports an interactive save-position mode, in addition to the normal auto sequence execution mode. In the save-position mode, the operator can interactively save the current arm position either in teleoperation or in operator-commanded auto-move mode and generate motion commands that can be used later in writing auto-sequence scripts. In the auto sequence execution mode the auto sequence script is displayed in the scrolled list window (Fig. 5; left middle window in Fig. 3), and the current command being executed is highlighted. The operator can interrupt the execution at any time by clicking the “pause” button, and later resume it by clicking the “run” button. The operator can also abort the remaining execution completely by clicking the “abort” button. The robot motion auto sequence commands include `rbt_move`, `rbt_pause`, `rbt_speed`, and `rbt_tool_length`. The image inspection auto sequence commands include image operations (e.g., `img_freeze`, `img_display`, `img_subtract`), light control, and video switch commands (camera control commands are not currently available, since the computerized camera control hardware is not yet installed). Note that the auto sequence script of Fig. 5 contains both robot motion and image inspection commands.

The on-line graphics display showing the current robot arm configuration at any desired viewing angle (upper right window in Fig. 3) is also provided as an effective visualization aid [9], [10] to the operator for more comfortable and safer operations. The actual robot joint angles are received from the remote site through a socket communication, and then used to update the graphics. When the task environment is known, the robot arm visualization can be extended to the task visualization by showing the operator a graphical simulation of the entire task environment including the robot arm.

3.2 Human Visual Inspection with Automated Scanning

The automated scanning capability has been introduced to the remote surface inspection system to reduce the operator’s manipulation workload and thus allow the operator to concentrate on inspection during the human visual inspection. Although a simple auto-scan function is in fact available in many commercial camera pan-tilt units for surveillance, our implementation provides more sophisticated flexible automated scanning capabilities based on the automated scanning GUI supporting operator-interactive sensor planning, scan path preview, and on-line graphical visualization (pictorial presentation) during the actual automated scanning.

The automated scanning procedure (Fig. 6) is conveniently divided into two phases: sensor planning and execution phases. In the sensor planning phase, the operator can register the object image and generate a scan path interactively. For the object registration, a pop-up window (Fig. 7) is used. Currently the object surface to be inspected is assumed to be approximately a 2-D rectangular surface, although it can be extended to other surfaces such as irregular 2-D shapes or cylindrically curved surfaces. In order to register the object image and generate a scan path for a rectangular surface, the operator moves the inspection camera by teleoperation to each corner of the rectangular surface so that the video image of the corner point appears at the center of the video monitor screen (a cross hair mark at the center of the monitor screen can be helpful), while keeping the inspection camera at a desired distance from the surface. Each corner point is registered by reading the camera position and marking the corresponding corner image point in the far-view window. If a far-view object image for flaw marking is not available, the far-view window simply displays a wire-frame rectangle.

After the registration, a scan path is generated so that the inspection will take place to cover the entire rectangular surface for a given camera view size. Two default scan paths are available: the horizontal path which scans the surface horizontally starting from the top row to the bottom and the vertical path which scans the surface vertically from the left column to the right. An example of a default horizontal scan path is shown in Fig. 8. For effective visualization, the scan path graphics is directly overlaid on the far-view digitized video object image. Along the scan path, via points (small squares in Fig. 8) for the robot motion trajectory and vista points (large squares in Fig. 8) where automated inspection takes place are defined. The vista points are placed so that the automated inspection will cover the entire surface. All vista points are defined as via points, but additional via points are inserted to ensure smooth and accurate scanning.

The operator can preview the generated scan path. During the preview, the operator can modify the scan path interactively by inserting, deleting, or relocating vista and via points as needed. This scan-path editing capability allows the operator to interactively generate a scan path for irregular 2-D shapes other than the rectangular shape. Once the image registration and the scan path generation are completed, the relevant image registration and scan path data are saved so that these procedures can be skipped for the subsequent inspections.

When the operator initiates scanning in the auto-scan execution phase (Fig. 6), the arm

first moves to the nearest via point from the current arm position, and then follows the scan path in the specified direction which can be either forward or backward. The operator can also request the arm to go to a designated via point directly. In addition to the scan path graphics, a rectangular box indicating the current position and the view port of the inspection camera is also overlaid on the far-view image (Fig. 8). As the robot arm carrying the inspection camera moves, the camera viewport rectangular box moves accordingly.

3.3 Machine-Vision-Based Automated Inspection

The operator's workload and inspection time can be significantly reduced by providing the operator with an automated inspection capability. By relying on the automated inspection, the operator does not have to inspect the entire surface, but examines a few portions of the inspection surface only when the automated inspection notifies the operator of a potential flaw. The operator ultimately decides whether there is a flaw on the surface. The JPL remote surface inspection system uses the machine-vision-based image differencing technique incorporated with automated scanning to provide the automated inspection capability. By selecting the "reference scan" button, the operator requests the inspection system to scan the object surface along the pre-defined scan path and collect a reference or "before" image for each vista point. Then, after a period of time, the operator selects the "comparison scan" button, requesting the inspection system to re-scan the surface along the scan path identical to the one used in the reference scan. During the comparison scan, the comparison or "after" image is compared with the reference or "before" image for each vista point. A large discrepancy between the two images implies a potential flaw occurrence.

In on-orbit space applications such as in Space Station Freedom surface inspection, the ambient lighting condition continuously varies due to the constant change of the sun illumination angle. When the reference and comparison images are taken under different ambient lighting conditions, a simple image differencing will generate too many false alarms, notably due to the shade change. A simple solution to this problem, although not practical, is to block the ambient lighting completely and use only the controlled lights mounted on the robot arm end effector. A good practical solution that achieves the same above effect has been developed and implemented in the JPL remote surface inspection system [3], [4]. First the surface image is taken with the controlled lights on, and immediately thereafter the image is taken again with the lights off. The differencing of the two images results in a compensated image as if there were no ambient lighting. This technique is thus used to

obtain the compensated “controlled-light-only” reference and comparison images, and these compensated images, not the raw images, are actually used in detecting the changes of the surface by image differencing.

When a large discrepancy is detected between the compensated reference and comparison images, the inspection system interrupts the automated scanning and notifies the operator of a potential flaw occurrence for further examination. First the system shows the operator the thresholded binary image obtained after the image differencing in the close-view window (Fig. 9a). The blob in Fig. 9a is due to the missing screw. When the operator hits the return key after viewing the image differencing result, the system shows the operator the compensated comparison or “after” image on which the flaw outline graphics is overlaid (Fig. 9b). This helps the operator to locate the flaw more quickly. The operator can now examine the flaw carefully by observing various available reference and comparison images of that flaw including those from previous scans. For this examination, the operator can use the on-line flaw manipulation capability described in the next section. Of course, the operator can use the teleoperation mode to move the arm around and observe the flaw region more carefully at different viewing angles and different controlled lights intensity levels. After the careful examination, the operator can either confirm the flaw occurrence and log the flaw in a data base, or ignore it if it is a false alarm. The operator can resume the automated inspection after taking care of the flaw notification.

4 On-Line Flaw Manipulation

The integrated operator interface design also provides on-line flaw manipulation [17] for remote surface inspection. The operator can interactively mark flaw locations, annotate flaws, and save flaw images. The operator can also retrieve and view flaw location maps, flaw annotations, and flaw image displays.

When the operator sees a flaw in the video image captured on the close-view window, the operator can enter the flaw location by marking its location with a mouse directly of the far-view image in the far-view window. To further facilitate the flaw marking, a “zoom in” capability is provided. When the “zoom in” function is selected, the portion of the far-view image enclosed by the current rectangular box indicating the current viewport of the inspection camera is enlarged to cover the entire far-view window. Except the reduced resolution, the “zoom in” image of the far-view window should be very similar to the close-

view image of the close-view window, and thus facilitating the flaw marking.

The operator can also extract and save the flaw image of a small rectangular region by clicking the flaw in the close-view window with a mouse. After the flaw image is saved in a flaw data base, a flaw annotation window (Fig. 10) pops up so that the operator can optionally enter commentary notes for the flaw image.

The flaw history table (Fig. 11) has been developed to allow the operator to efficiently review flaw images from previous scans, by noting that the same scan path will be used repeatedly for a given object. Each column in the table, labeled with the date of the scan, contains the flaws of that scan. Each row, labeled with the flaw ID, contains the flaws of the same or near same locations in different scans. Each entry in the table shows the location of the flaw. The operator can double-click on an entry to view the image of that particular flaw in the close-view window. When the operator double-clicks on a date of scan (column label), all the flaws marked in that scan are displayed in the close-view window. When the operator double-click on a flaw ID (row label), all the flaw with that ID are displayed (Fig. 12). This allows the operator to observe the chronological change of a particular flaw. For example, Fig. 12 indicates that the screw was in only on October 20 in this artificial scenario. The annotation of a particular flaw (Fig. 10) can be brought up by double-clicking its image displayed in the close-view window.

5 Conclusion and Future Plans

Integrated graphics and graphical user interfaces were effectively used in the new development of an efficient flexible graphical operator interface design for a telerobotic inspection system. The operator interface was designed so that remote inspection can be performed by a single operator with an integrated manipulator control and image inspection capability. The interface supports three inspection strategies of teleoperated human visual inspection, human visual inspection with automated-scanning, and machine-vision-based automated inspection.

Future plans include 1) extension of the sensor planning capability from the current 2-D and the 3-D scan path generation, 2) employment of other sensors in addition to visual inspection cameras, and 3) experiments/evaluation to quantify performance enhancements with the newly developed operator interface design.

Acknowledgment

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

References

- [1] R. M. Baecker and W. A. S. Buxton, *Readings in Human-Computer Interaction*, Morgan-Kaufmann Publishers, Los Altos, CA, 1987.
- [2] F. Fisher and C. R. Price, "Space Station Freedom External Maintenance Task Team Final Report," NASA Johnson Space Center, Houston, TX, Jul. 190.
- [3] S. Hayati, J. Balaram, H. Seraji, W.S. Kim, K. Tso, and V. Prasad, "Remote Surface Inspection System," IEEE Conf. on Robotics and Automation, pp. 875-882, Atlanta, GA, May 1993.
- [4] S. Hayati, H. Seraji, W. S. Kim, and J. Balaram, "Remote Surface Inspection for Space Applications," *Proc. AIAA Aerospace Design Conf.*, AIAA 92-1017, Irvine, CA, Feb. 1992.
- [5] F. Hayes and N. Baran, "A Guide to GUIs," *Byte*, vol. 14, no. 7, pp. 250-257, July 1989.
- [6] Johnson Space Center, "Shuttle Flight Operations Manual: vol. 16 — Payload Development and Retrieval System," NASA Johnson Space Center, Houston, TX, 1981.
- [7] Johnson Space Center, "Space Station Freedom Program Human-Computer Interface Guide," USE 1000 version 2.1, NASA Johnson Space Center, Houston, TX, 1988.
- [8] W. S. Kim , "Graphical Operator Interface for Space Telerobotics," *IEEE Conf. on Robotics and Automation*, pp. 761-768, Atlanta, GA, May. 1993.
- [9] W. S. Kim and A. K. Bejczy, "Graphics Displays for Operator Aid in Telemanipulation," *IEEE Conf. on Systems, Man, and Cybernetics*, Charlottesville, VA, Oct. 1991.
- [10] W. S. Kim, P. Schenker, and A. K. Bejczy, "Graphics Simulation and Training Aids for Advanced Teleoperation," *Sixth Annual Symposium on Space Operations, Applications, and Research (SOAR '92)*, NASA Johnson Space Center, Houston, TX, Aug. 1992.
- [11] P. Lee, A. K. Bejczy, P. S. Schenker, and B. Hannaford, "Telerobot Configuration Editor," IEEE Int. Conf. on Systems, Man, and Cybernetics, Los Angeles, CA, 1990.

- [12] Open Software Foundation, *OSF/Motif Programmer's Reference*, release 1.1, 1991.
- [13] T. Rubin, *User Interface Design for Computer Systems*, Ellis Horwood Ltd., West Sussex, England, 1988.
- [14] T. See *et al.*, "Meteoroid and Debris Impact Features Documented on the Long Duration Exposure Facility: A Preliminary Report," Space and Life Science Branch Publication #84, JSC #24608, Aug. 1990.
- [15] D. C. Smith, C. Irby, R. Kimball, W. Verplank, and E. Harslem, "Designing the Star User Interface," *Byte*, vol. 7, no. 4, pp. 242-282, April 1982.
- [16] D. E. Smyth, "Wcl — Widget Creation Library, Mri — Motif Resource Interpreter, Ari — Athena Resource Interpreter: An Easier Way to Develop Applications using Motif and Athena Widgets," *Proc. of the European X Window System User Group Autumn Conference*, 1990.
- [17] K. S. Tso, "A Fault Tolerant Intelligent Robotic Control System," SBIR Phase-I Final Report under NASA Contract NAS7-1172, SoHaR Incorporated, Beverly Hills, CA, Sept. 1992.
- [18] D. A. Young, *The X Window System Programming and Applications with Xt: OSF/Motif Edition*, Prentice Hall, 1990.

Figure Legends

Figure 1. JPL telerobotic inspection system with a 1/3 scale Space Station Freedom truss structure, ORU's, and a robot arm carrying inspection cameras and lights

Figure 2. Operator control station housed in a realistic cupola mockup of the Space Station Freedom

Figure 3. Top-level screen layout of the graphical operator interface for the JPL telerobotic inspection system.

Figure 4. Robot control GUI. (a) top-level, (b) "hand controller scale" pull-down menu, and (c) "set scale" pop-up window.

Figure 5. Auto sequence GUI

Figure 6. Schematic of the automated scanning procedure

Figure 7. Object registration pop-up window

Figure 8. Automated scanning GUI with graphics overlay of a default horizontal scan path

Figure 9. Automated inspection notifies the operator of a potential flaw occurrence first with (a) the thresholded binary image after image differencing and then with (b) the flaw-outline-overlaid compensated comparison image. The actual flaw is the missing screw.

Figure 10. Flaw annotation text window

Figure 11. Flaw history table

Figure 12. Flaw images display

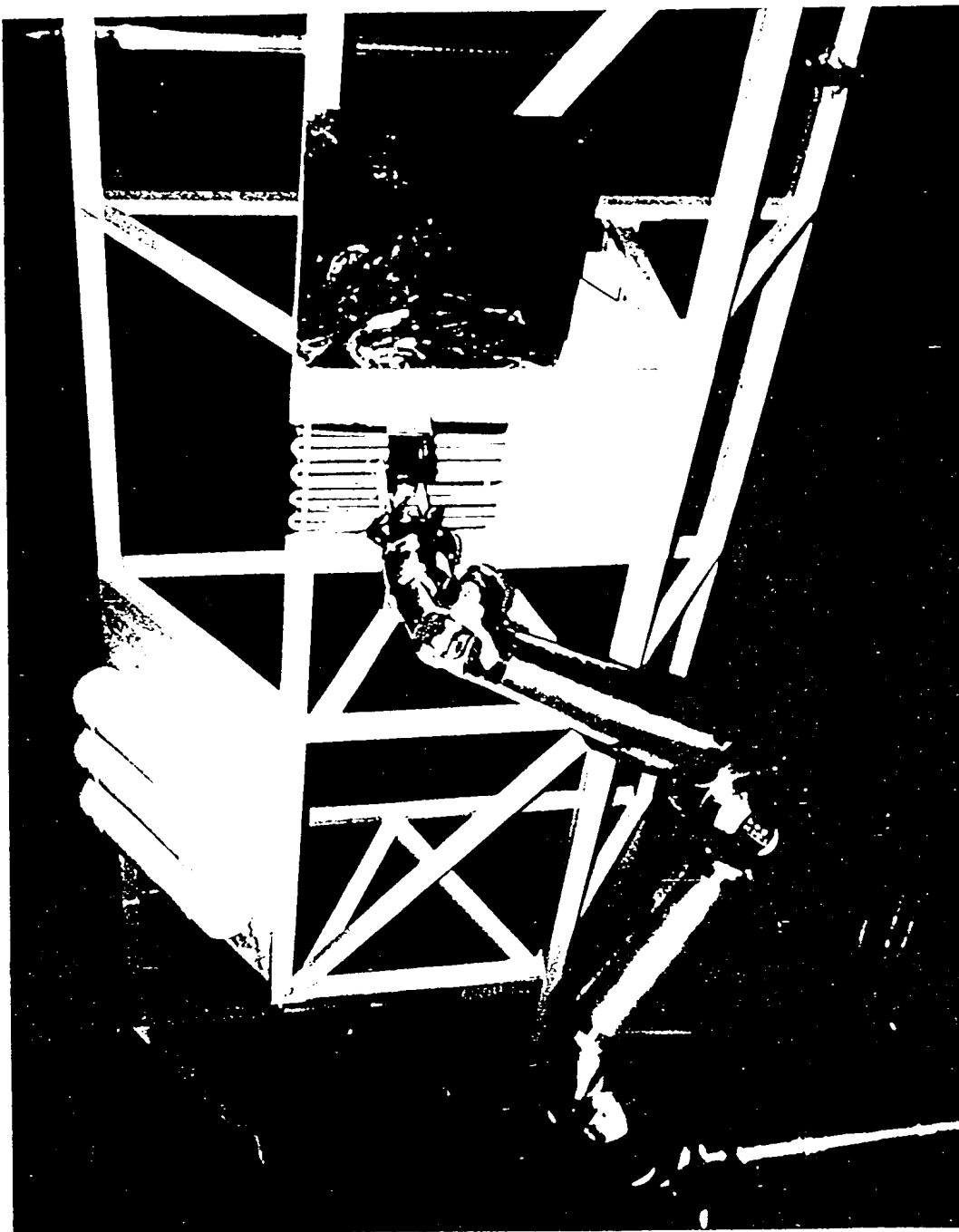


Fig. 1

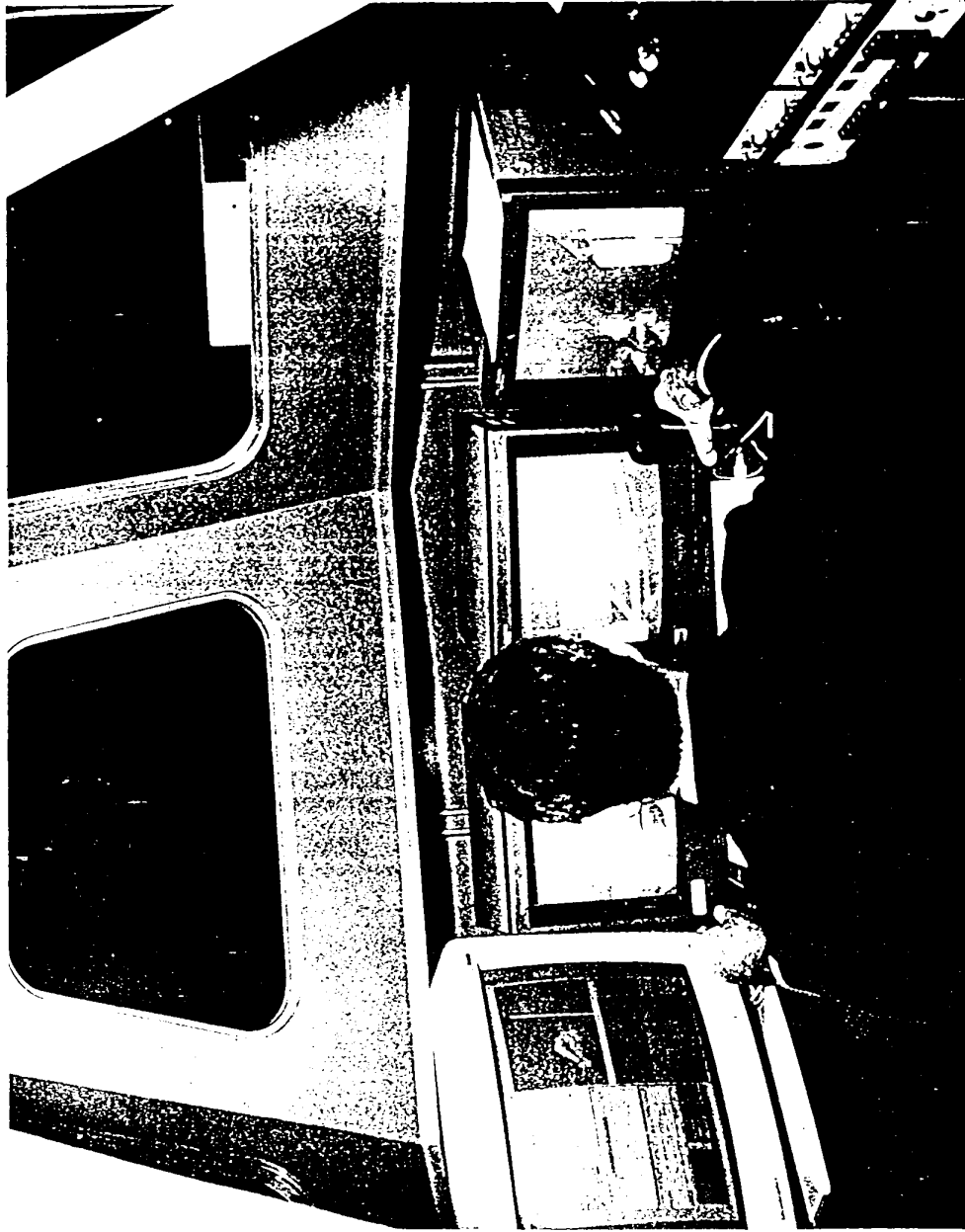


Fig. 2

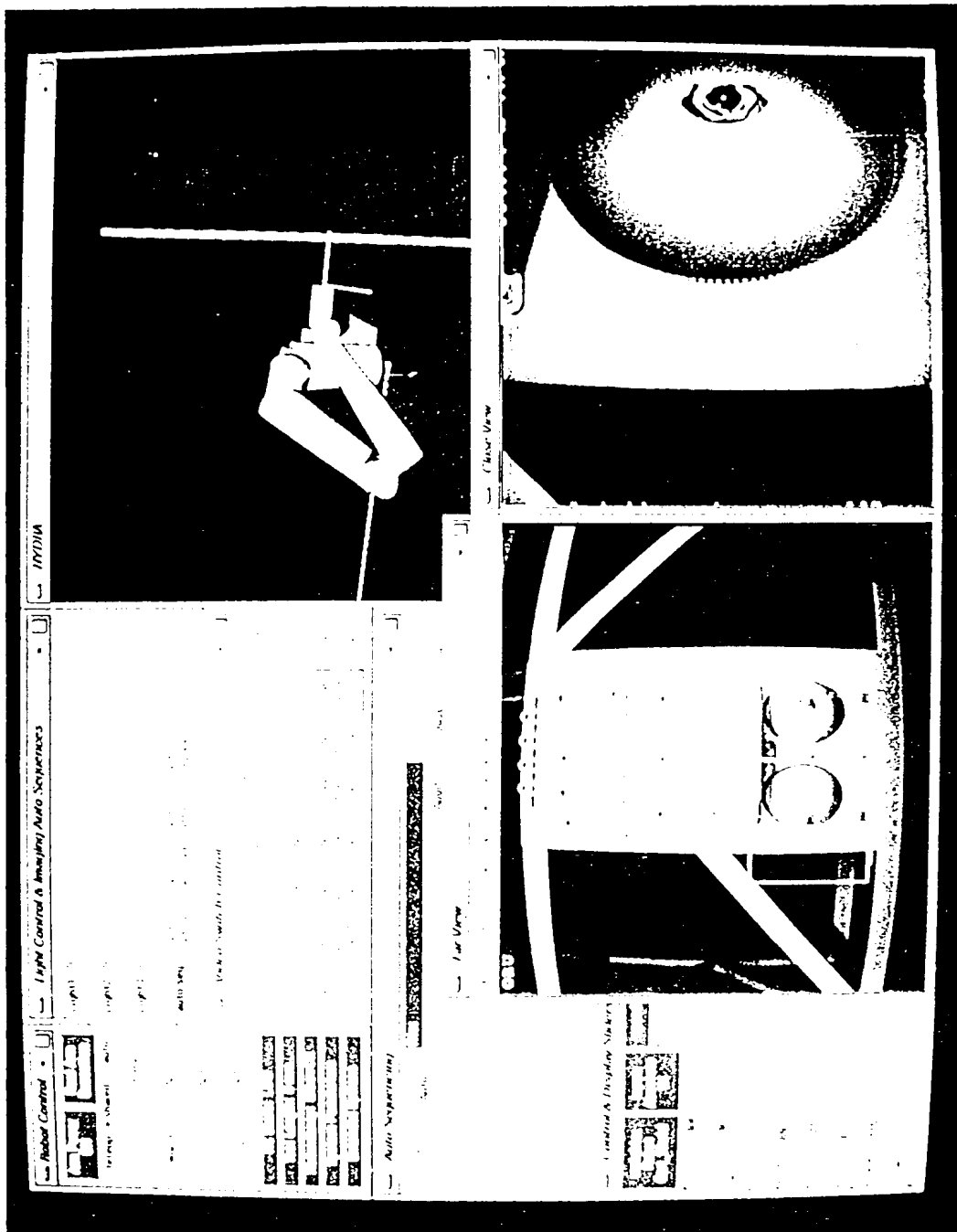


Fig. 3

5

year ccd n

high freq



1. The first step is to identify the problem or question that needs to be answered. This involves understanding the context and the specific requirements of the task.

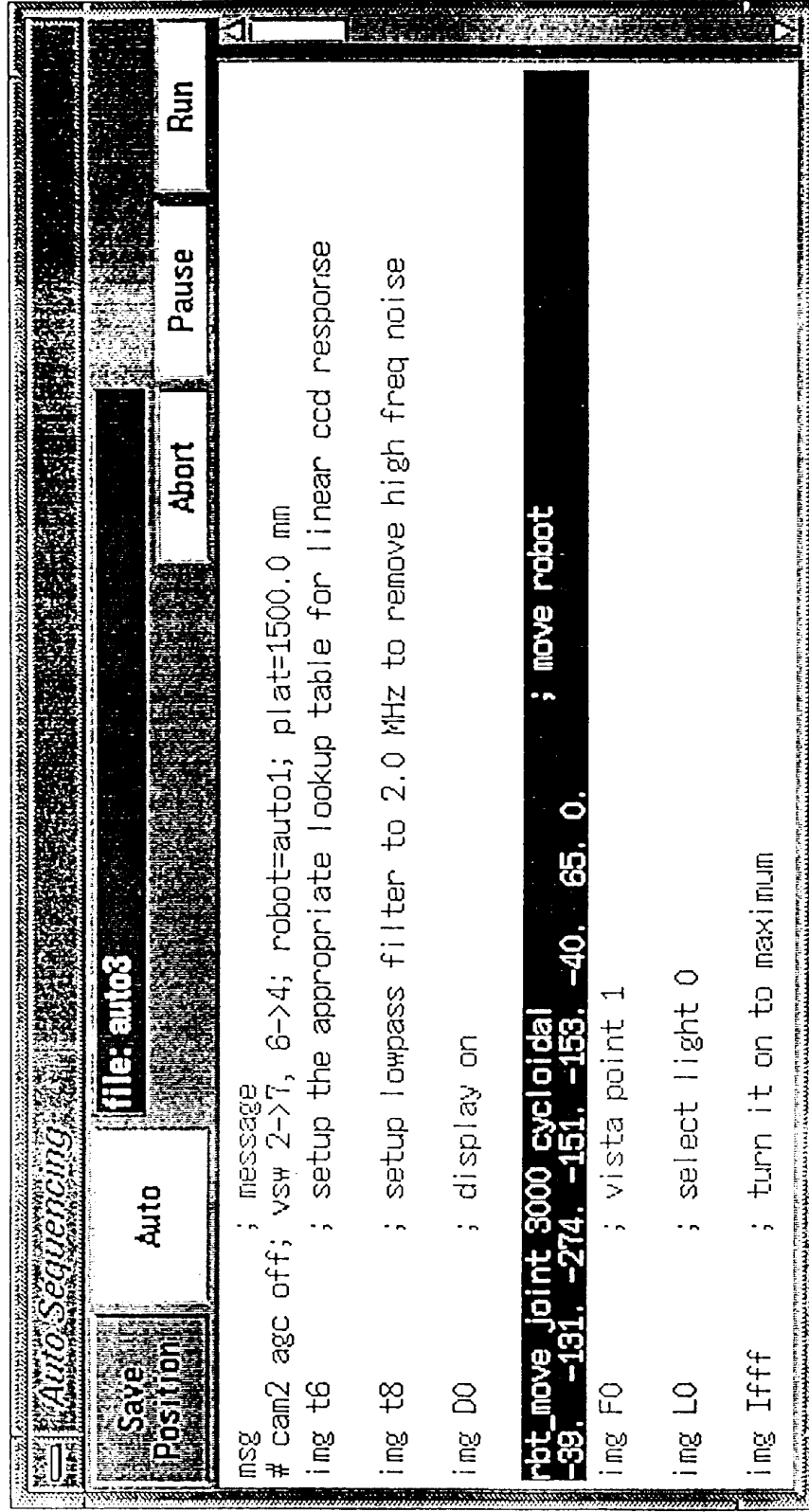


Fig. 5

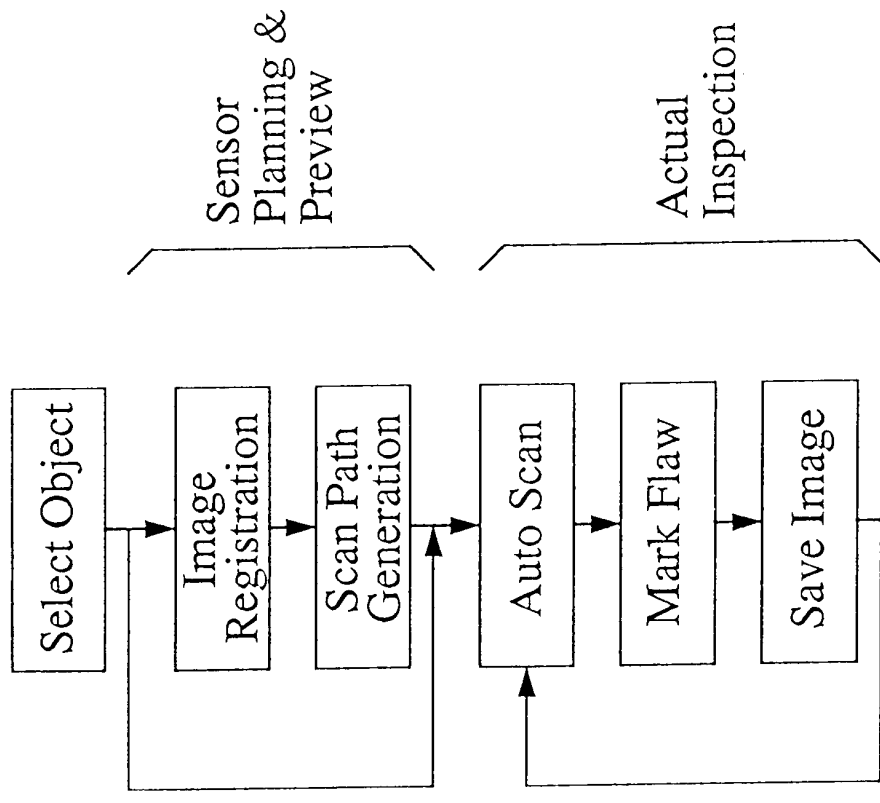


Fig. 6

Object Registration

Name:

ORU

View Size:

75.00

Image Position

Camera Position

Id	X	Y	X	Y	Z
1	158	24	324	-514	-461
2	355	29	767	-520	-461
3	358	447	767	206	-461
4	150	444	316	213	-461

Done

Image

Camera

Cancel

Fig. 7

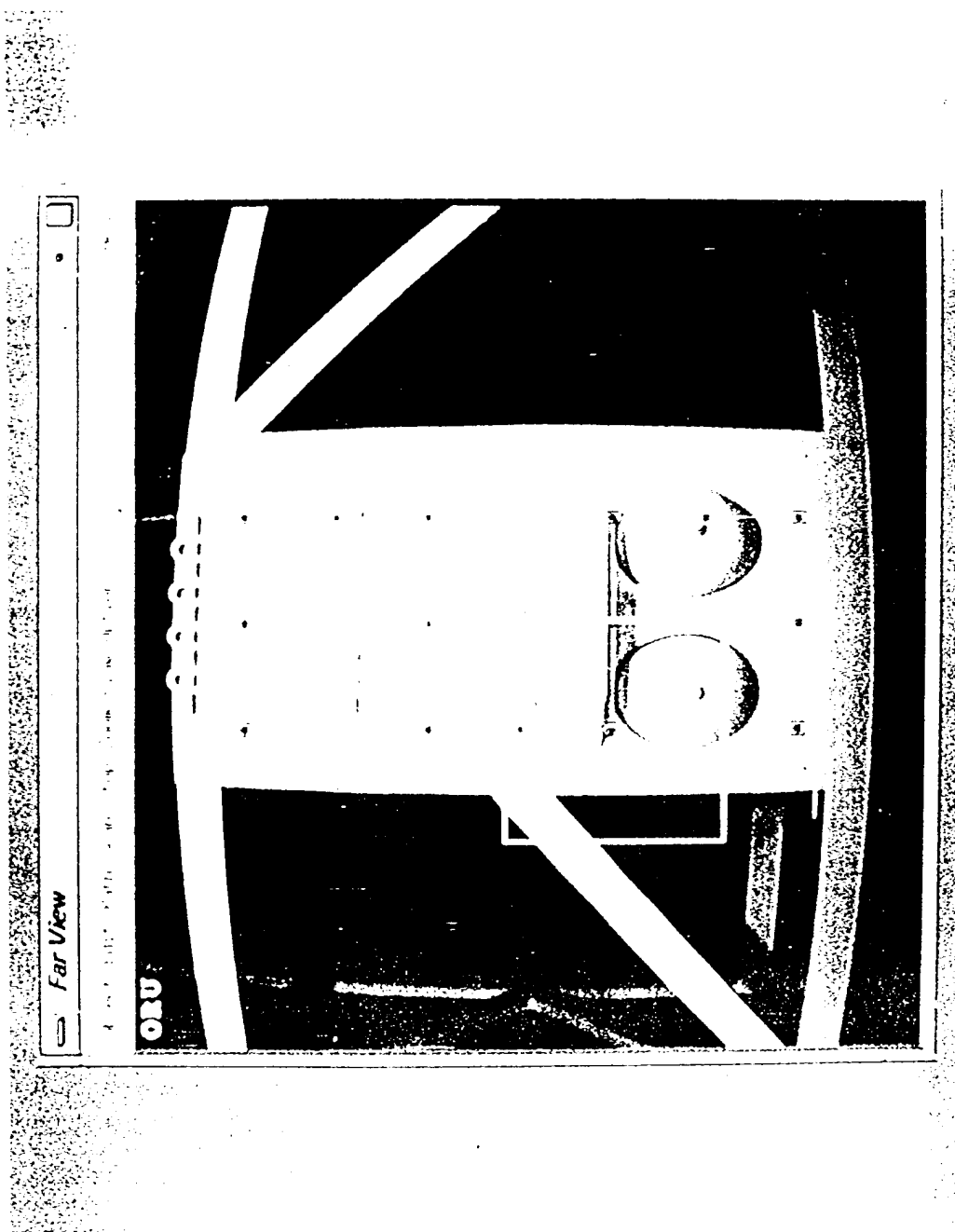
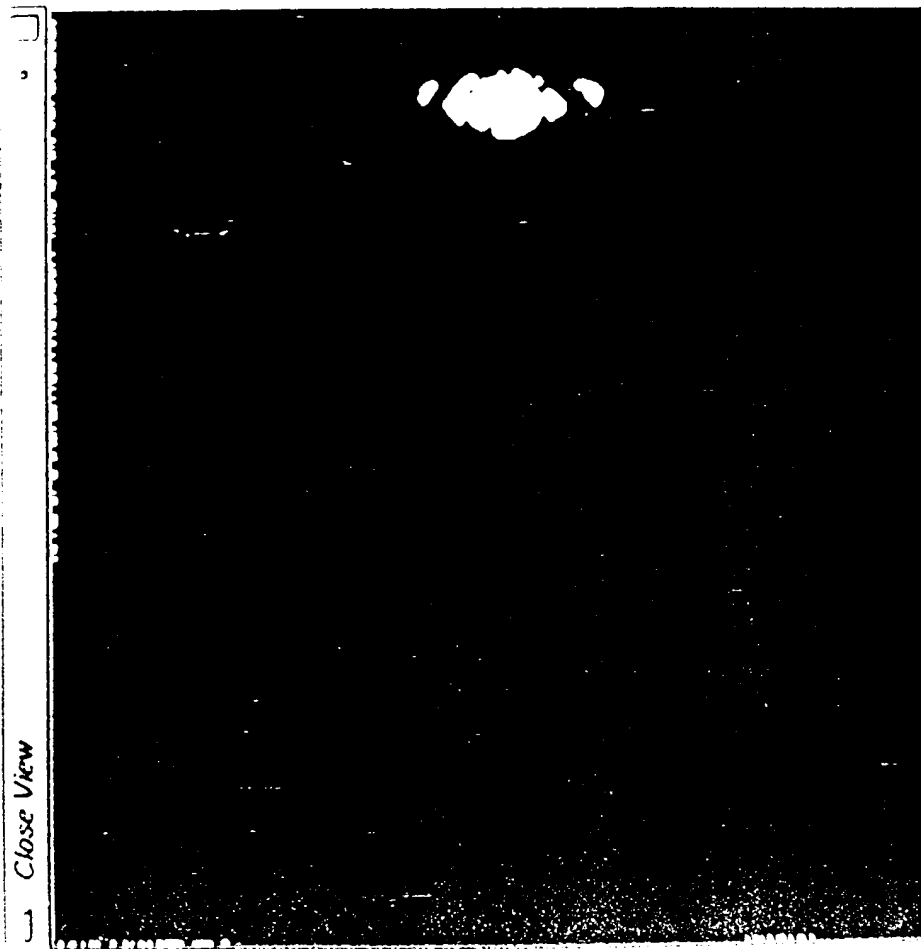
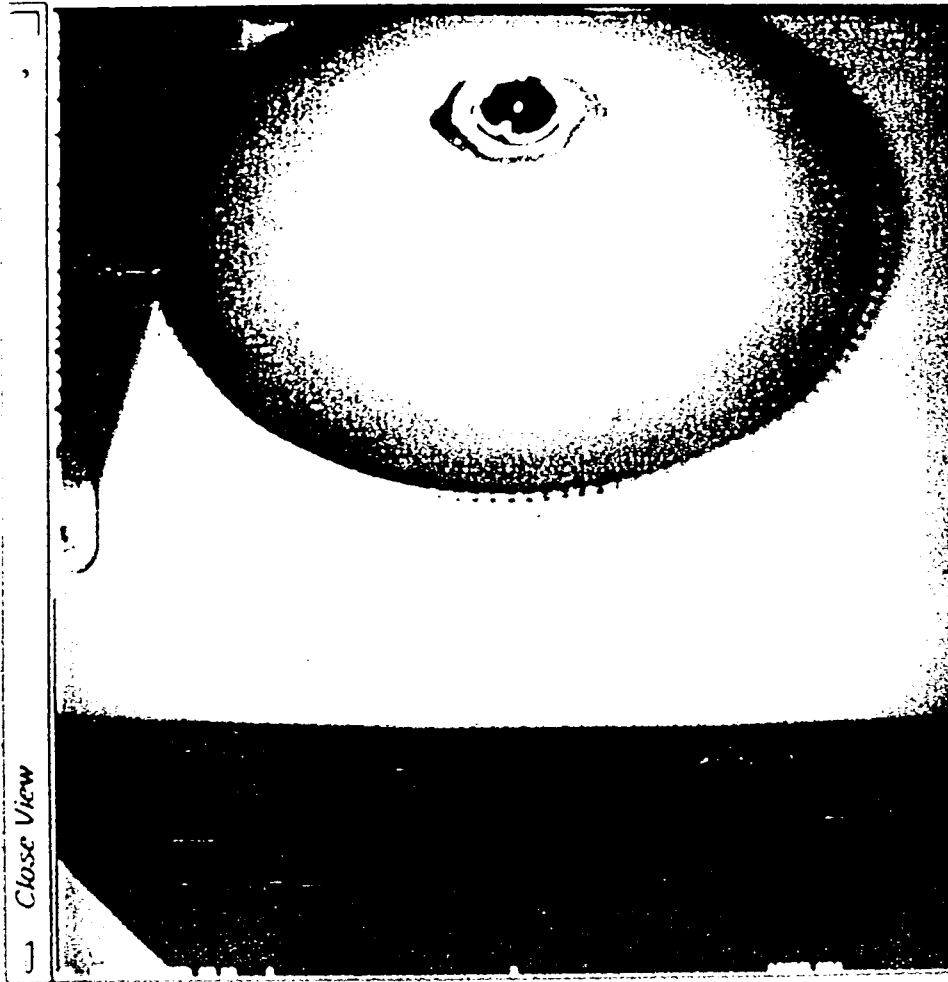


Fig. 8



Close View

Fig. 9 (a)



Close View

Fig. 9 (b)

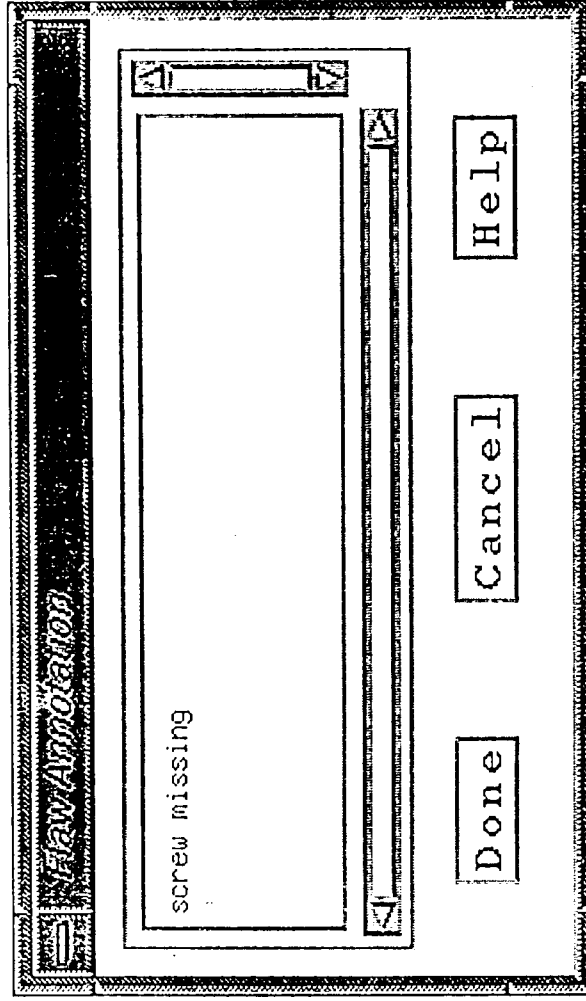


Fig. 10

Flaw History				
Flaw ID	Oct 24 1993	Oct 23 1993	Oct 23 1993	Oct 23 1993
1	(242, 342)	(242, 341)		
2	(326, 136)	(326, 135)		
3	(210, 139)	(209, 138)		
4			(321, 152)	(323, 152)
5			(204, 152)	(203, 154)
6			(226, 348)	(224, 348)

Fig. 11



Fig. 12